

## 55. IWK

Internationales Wissenschaftliches Kolloquium  
International Scientific Colloquium



13 - 17 September 2010

# Crossing Borders within the **ABC**

## **A**utomation,

## **B**iomedical Engineering and

## **C**omputer Science



Faculty of  
Computer Science and Automation

[www.tu-ilmenau.de](http://www.tu-ilmenau.de)

*th*  
TECHNISCHE UNIVERSITÄT  
ILMENAU

Home / Index:

<http://www.db-thueringen.de/servlets/DocumentServlet?id=16739>

## **Impressum Published by**

Publisher: Rector of the Ilmenau University of Technology  
Univ.-Prof. Dr. rer. nat. habil. Dr. h. c. Prof. h. c. Peter Scharff

Editor: Marketing Department (Phone: +49 3677 69-2520)  
Andrea Schneider (conferences@tu-ilmenau.de)

Faculty of Computer Science and Automation  
(Phone: +49 3677 69-2860)  
Univ.-Prof. Dr.-Ing. habil. Jens Haueisen

Editorial Deadline: 20. August 2010

Implementation: Ilmenau University of Technology  
Felix Böckelmann  
Philipp Schmidt

## **USB-Flash-Version.**

Publishing House: Verlag ISLE, Betriebsstätte des ISLE e.V.  
Werner-von-Siemens-Str. 16  
98693 Ilmenau

Production: CDA Datenträger Albrechts GmbH, 98529 Suhl/Albrechts

Order trough: Marketing Department (+49 3677 69-2520)  
Andrea Schneider (conferences@tu-ilmenau.de)

ISBN: 978-3-938843-53-6 (USB-Flash Version)

## **Online-Version:**

Publisher: Universitätsbibliothek Ilmenau  
[ilmedia](#)  
Postfach 10 05 65  
98684 Ilmenau

© Ilmenau University of Technology (Thür.) 2010

The content of the USB-Flash and online-documents are copyright protected by law.  
Der Inhalt des USB-Flash und die Online-Dokumente sind urheberrechtlich geschützt.

## **Home / Index:**

<http://www.db-thueringen.de/servlets/DocumentServlet?id=16739>

# A METHOD OF NONLINEAR SYSTEM OPTIMIZATION BASED ON HYBRID EVOLUTIONARY COMPUTATION TECHNIQUES AND ITS APPLICATION FOR IMAGE CLASSIFICATION

Uwe Fohry

Fachhochschule Jena, Carl-Zeiss-Promenade 2, 07745 Jena

## ABSTRACT

This paper presents a general method using a hybrid evolutionary search strategy to support design and optimization of nonlinear systems. Depending on the task like system identification, classification or process control a defined quality criterion of the system is gradually improved in an experimental manner. The optimization variables consist of chosen parameters and structure elements of the system. For that a global search method is used, that incorporates gradient-based optimization algorithms. The aim of the hybridization is to improve the efficiency in local finetuning, while maintaining the global search behaviour. In the second part the method is specified for the application of image classification. It is explained, how the method was used for aliveness detection of fingerprints.

**Index Terms** – Evolutionary Computation, classification, image recognition

## 1. INTRODUCTION

Design and optimization of a system require the determination of variables to achieve a sufficient optimal system behavior in terms of a defined performance. To support these tasks by the use of computer assistance, it may be considered as optimization algorithm, shown in Figure 1. An in general nonlinear dynamic system is available as simulation model, as model approach or as real system. The variables of the system are combined to the decision vector  $\underline{v}$ . That may contain internal parameters of the system, structure determining variables as well as control parameters of strategies. The system behaviour  $\underline{x}_a(t) = S[\underline{x}_e(t), \underline{v}]$  can be calculated and evaluated by suitable excitation. This occurs according to determined criteria by a user-defined quality functional  $E\{\underline{x}_a(t)\}$ . Commonly  $E$  represents a scalar error value. Therefore the objective function is  $E(\underline{v})$  and the optimization problem may be seen as minimization problem

$$\min_{\underline{v} \in \mathbb{N}} E(\underline{v}), \text{ where } \mathbb{N} = \{\underline{v} \mid \underline{g}(\underline{v}) = \underline{0}; \underline{h}(\underline{v}) \leq \underline{0}\}. \quad (1)$$

The permitted range  $\mathbb{N}$  of the decision variables may be restricted by side conditions.

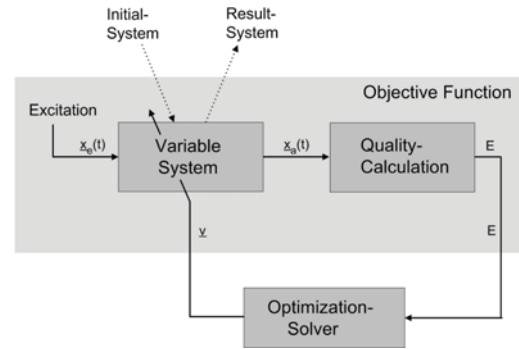


Figure 1 Principle of System Optimization

Problem (1) can be solved by an optimization method either in an analytic manner or based on a given initial system in an algorithmic manner. In this matter efficiency and application requirements are generally opposed features of the methods. Mathematical methods of nonlinear optimization offer a comparative high speed of convergence and require gradient and possibly the Hessian of the objective function. For this purpose the objective function should be available as differentiable mathematical expression, what often can not be done. An alternative for that is to use derivative-free local search methods. This way allows an experimental determination of the system error during the optimization. Both gradient-based optimization algorithms and derivative-free local search methods are only able to find a local minimum depending on the initial solution. Therefore these local methods are not suitable for topographical complex objective functions containing local suboptima. An important tool in the category of global search methods are provided by the Evolutionary Algorithms (EA), which represent a basis for this work. Their applicability for system optimization be discussed below.

Evolutionary algorithms denote a summarization of heuristic search methods, which are motivated from principles of natural evolution. Based on abstract emulation of genetic variation and fitness depending selection in an iterative manner these algorithms represent universal solver for various optimization problems. In respect of their algorithmic

subgroups, descriptions of functionality and practical applicability be referred to the numerous literature, i.e. [1]. EA can be used for design and optimization of systems. There a system is seen as a individual and the evolutionary loop operates on a whole population of proposed systems, which are different in the determinable variables. Summarized there are following advantages of the evolutionary system optimization compared to classical methods:

- A mathematical formulation and differentiation of the objective function is not necessary.
- Thus the quality of the system may be detected experimentally during the optimization.
- Available expert knowledge can well be included by initial proposals of solutions.
- The objective function may be multi-modal because of the global search behaviour.
- The objective function may be continuous as well as discontinuous.
- It is possible, to find a pareto-optimal solution set according to a multi-criterial system quality.
- Because of the population-based procedure a good parallelizing of the algorithm is possible.
- Within an EA can be used several strategies simultaneously.
- EA can be hybridized by integration of alternative methods.

These advantages are accompanied by the following disadvantages:

- The detection of a local error minimum with sufficient precision requires a good many system evaluations.
- EA possess a lot of possible operations and control parameters.

To improve the practical applicability of the evolutionary system optimization, the advancement of algorithms focus on techniques to increase the convergence. For this purpose the following proposed algorithm uses the possibility of hybridization. Thereby chapter 2 relates to the optimization of continuous objective functions by hybridization of an Evolution Strategy. In chapter 3 this method is incorporated into an algorithmic concept including the search for a fitting structure of the system.

## 2. PARAMETER-OPTIMIZATION USING A HYBRID EVOLUTION STRATEGY

The noted disadvantages of convergence of EA are referred especially to the optimization of real parameters in comparison with local optimization and search methods. Therefore the described algorithm relates to the optimization of static system parameters  $\underline{p} \in R^n$ . In several researches as well as preliminary experiments could be verified significant improvements in performance by the implementation of local search methods, i.e. [2]. The resulting hybrid EA, so called memetic algorithm, organizes the global search behaviour so, that the exploration of the search

space is done by the evolutionary operators, which is noted at their parametrization. On the other hand the exploitation, that means the stepwise improvement of an established solution, is the task of the implemented local method.

In preliminary investigations was detected, that for an evolutionary optimization of real parameters especially Evolution Strategies (ES) are suitable. Therefore they are used at the suggested algorithm. In Figure 2 is shown the fundamental flow of an ES, at which the gray blocks mark the performed extensions for the implementation of local methods. As local methods can be chosen depending on the differentiability of the objective function either gradient-based or derivative-free methods. In the context of derivative-free methods the best experimental results were reached by using deterministical neighbourhood search algorithms like the method of Hooke-Jeeves [3].

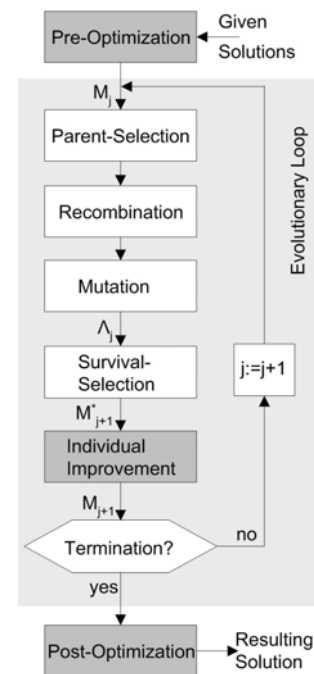


Figure 2 Evolutionary Search of Parameters

In the evolutionary loop at Figure 3  $M_j$  presents the set of existing proposals of solutions (individuals) at iteration (generation)  $j$ . Some of this individuals are selected as parents to generate new solutions  $A_j$  by recombination and mutation. A number of the best offsprings are taken over the next generation  $(j+1)$ . By the additional operator of individual improvement the best individual from  $M_{j+1}^*$  is undergone of a local method, if it significantly different from the current best solution. Thereby the exploitation of the search space are accelerated because of the directed search way of the local method.

The pre-optimization is used to generate the initial population. Thereby the aim is to fit the size and the diversity of the initial population to the complexity of

the objective function and further to get already good solutions there. That is striven in substance so, that several local searches are processed, where the startpoints are spaciouly spread into the searchspace. Results of these searches, of which the Euclidean distance fall below a determined threshold are summarized to one individual.

After the end of the evolutionary loop the best result can be stated more precisely by a local method within the post-optimization.

### 3. STRUCTURE SEARCH BY SUB-POPULATIONS

The proposed method can be incorporated into a whole algorithm Figure 3 consisting of a structure search and of a subordinated parameter search.

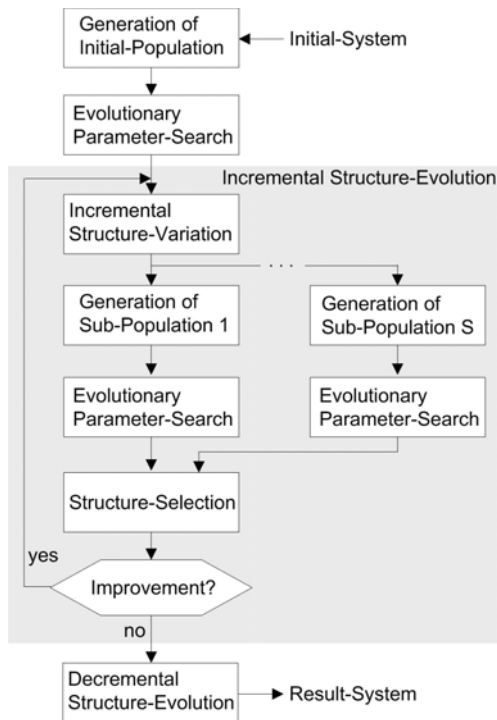


Figure 3 Whole Procedure

An essential feature of the algorithm is the gradual increasing of system complexity during the evolution. Structure oriented variables, as i.e. order of a polynomial, number of units in a neural network, number of chosen inputs etc., are decision variables from the range of the natural numbers (decision vector  $\underline{v} \in N^n$ ). Therefore an extension step inside of the incremental structure variation can be realized to several possibilities. Thereby it is produced  $S$  sub-populations, which are parallel undergone of a parameter search. The best result of these parameter searches is taken over the next generation by the structure selection. The incremental structure evolution is performed, so long as a structure extension step effects a significant improvement of the system behaviour. The following decremental

structure evolution allows also to find simpler structures as that of the initial system.

### 4. USING THE CONCEPT FOR IMAGE CLASSIFICATION

The presented concept of evolutionary system optimization are specified now to solve a classification problem on the example of image recognition. The concrete task is related to the classification of fingerprints, which are generated by a laser scanner. These objects should be referred to the class of living fingers or the class of imitations. For that are available some thousand examples of images and their right allocation. However it hardly exists expert knowledge about a mapping function of any features to the classes. Also that can be different among the individual scan devices. To avoid time-consuming studies, it would be well, to perform the extraction of the relevant features as well as the determination of the structure and the parameters of the classifier in a mainly automated experimental process. This is illustrated by Figure 4 and may be regarded as a specialization of the principle in Figure 1. It should be noted, that the explanations in the following chapters also apply to similar tasks in the range of databased supervised learning.

By operators of image processing are features provided based on gray-level values and texture of the image. The concrete features are not named because of confidentiality. From this supplied features can be selected the relevant features for the classification. By a functional relation, following referred to as a classifier, are calculated out of it a real estimation value. After the binarization by means of an user defined threshold are issued either 1 for a living object or -1 for a imitation. Depending on the examples are calculated a classification error  $E$  by comparison between the estimated and the right allocation. The decision vector of the optimization is  $\underline{v} = (\underline{p}, \underline{s}, \underline{m})^T$ . The parameters  $\underline{p}$  of the classifier can be estimated by the algorithm of parameter search at chapter 2. The structure parameters  $\underline{s}$  of the classifier as well as the feature selection  $\underline{m}$  are relevant for the whole procedure at chapter 3.

In preliminary experiments are experienced, that a linear separation of the objects in the supplied feature space does not achieve a respectable result. Therefore are chosen a nonlinear classifier. Due to the relative high number of features should be used a classifier, where the number of free parameters depends linear on the number of inputs. For the determination of its structure is no knowledge available. That has to take place by a *trial and error* process. Thus the classifier should offer a few number of diverse structure parameters. Because of these reasons it was decided to use a feed-forward neural network especially a Multilayer Perceptron (MLP).

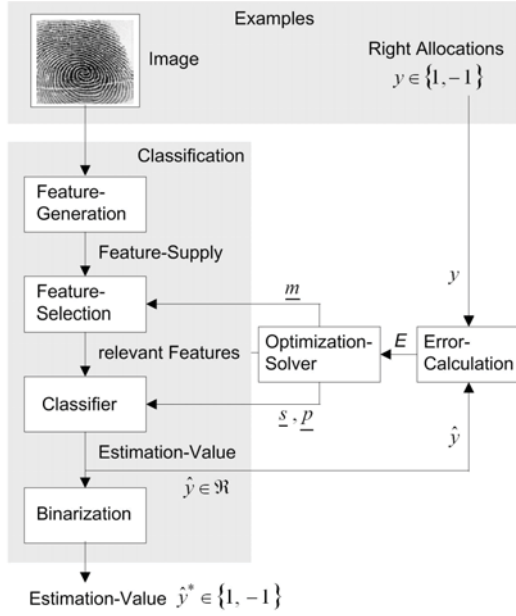


Figure 4 The Classification Problem

The specification of the in chapter 1 to 3 represented algorithm concept for this task are described following. Thereby at first it is considered in chapter 5 the parameter optimization in general view, to maintain the useability of the concept for arbitrary classifier. In chapter 6 the gained conclusions are used, while a MLP serves as classifier. It is explained, how the optimization can be extended by the search of relevant features and of a fit classifier structure.

## 5. OPTIMIZATION OF CLASSIFIER PARAMETERS

This chapter is applied to the estimation of the real parameters of the classifier using the data of examples. The optimization problem is formulated and characterised. It is deliberated, which solver method should be used and be implemented within the represented evolutionary whole algorithm. Thereby it is expected, that it was chosen a structure of classifier as well as relevant features in advance.

The classifier are provided by the functional relation  $\hat{y} = f(u_1, \dots, u_F, p_1, \dots, p_N)$ . Thereby  $F$  is the number of inputs, so the relevant features  $u_1$  to  $u_F$ . Also the function contains  $N$  free parameters summarized into the parameter vector  $\underline{p} = (p_1, \dots, p_N)^T$ . A data set is available by the examples in form of

$$\begin{array}{cccc|c} u_{11} & u_{12} & \cdots & u_{1F} & y_1 \\ u_{21} & u_{22} & \cdots & u_{2F} & y_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ u_{S1} & u_{S2} & \cdots & u_{SF} & y_S \end{array} \quad (2)$$

summarized into the input matrix  $\underline{U}$  and the target vector  $\underline{y}$ , whereas  $S$  is the number of right examples.

By means of these data should be done an offline estimation of the classifier parameters. The searched mapping function is not changing, so the classifier can seen as static time-invariant nonlinear system. The estimation error  $E$  is commonly calculated according to the *Least Square Method* (LS-Method). Thus is given the optimization problem as

$$\min_{\underline{p}} E(\underline{p}) = \frac{1}{2} \cdot \sum_{s=1}^S (\hat{y}_s(\underline{p}) - y_s)^2. \quad (3)$$

Hereby is represented a nonlinear optimization problem of static parameters. The nonlinearity is already generated by the LS-Method. There are no side conditions, because the permitted range of parameters is unrestricted.

The selection of the solver depends on the structure of the classifier function. If it is continuously differentiable to its parameter, gradient-based optimization methods can be used, otherwise derivative-free methods are interesting. The further explanations are referred to the case of parameter differentiability, so that the gradient can be used. The partial derivatives of (3) to  $\underline{p}$  ensue according to the chain rule as

$$\frac{\partial E(\underline{p})}{\partial p_n} = \sum_{s=1}^S \frac{\partial \hat{y}_s(\underline{p})}{\partial p_n} \cdot (\hat{y}_s(\underline{p}) - y_s), \quad \forall n=1, \dots, N. \quad (4)$$

By summarization of the estimation values  $\hat{y}_s(\underline{p})$  into the estimation vector  $\hat{\underline{y}}(\underline{p})$  and notice of the definition of the Jacobian matrix

$$\hat{\underline{Y}}'(\underline{p}) = \frac{\partial \hat{\underline{y}}_s(\underline{p})}{\partial p_n}, \quad \forall s=1, \dots, S, \quad \forall n=1, \dots, N \quad (5)$$

the gradient of the objective function can be calculated by

$$\nabla E(\underline{p}) = \hat{\underline{Y}}'^T(\underline{p}) \cdot (\hat{\underline{y}}(\underline{p}) - \underline{y}). \quad (6)$$

At first be considered the case, that the classifier function is linear in its parameters. Thus it can be represented of a linear combination of element functions  $g_n(\underline{u})$ , so that the estimation values can be calculated by

$$\hat{y}_s = \sum_{n=1}^N g_n(\underline{u}_s) \cdot p_n, \quad s=1, \dots, S. \quad (7)$$

After substitution of the results of the element functions  $m_{sn} = g_n(\underline{u}_s)$  and summarization into a matrix  $\underline{M} = m_{sn}$  can (7) be expressed as  $\hat{\underline{y}}(\underline{p}) = \underline{M} \cdot \underline{p}$ . It is obviously, that the Jacobian matrix of this linear vector function is  $\hat{\underline{Y}}'(\underline{p}) = \underline{M}$ . Setting in (6) the error gradient is



$$\nabla E(\underline{p}) = \underline{M}^T \cdot (\underline{M} \cdot \underline{p} - \underline{y}) \quad (8)$$

That is linear in  $\underline{p}$ , so that the minimum of the error at  $\underline{p}^{\min}$  can be determined by analytic calculation of the stationary point according to  $\nabla E(\underline{p}^{\min}) = \underline{0}$ . After inserting of (8) is got the well known equation

$$\underline{p}^{\min} = (\underline{M}^T \underline{M})^{-1} \underline{M}^T \cdot \underline{y} = \underline{M}^+ \cdot \underline{y} \quad (9)$$

Thus in the parameter-linear case the parameter search in the whole procedure Figure 3 can be substitute by the analytic calculation (9).

If the classifier function is nonlinear in its parameters, numerical methods of nonlinear optimization can be used. In respect of a particularly description of following mentioned methods are referred to the various literature. If  $k$  the current iteration, so the next parameter vector is calculated by means of the negative error gradient by

$$\underline{p}(k+1) = \underline{p}(k) - \underline{\gamma}(k) \cdot \nabla E(\underline{p}(k)) \quad (10)$$

where  $\nabla E(\underline{p}(k))$  is calculated by (6). The possible methods are different in the matrix  $\underline{\gamma}(k)$ . In the simplest case it is a scalar matrix using by methods of deepest descent. In more complex methods of second order for its calculation is used the Hessian of the objective function. At this place be mentioned the Newton-Method, which have a quadratic speed of convergence by using the invers Hessian as  $\underline{\gamma}(k)$ . Thereby is condition, that  $E(\underline{p}(k))$  is positiv semidefinite.

An expedient possibility to associate the high speed of the Newton-Method and the robustness of deepest descent is the method of Marquardt-Levenberg. It is based on a extension of the Newton-Method by a modification of the Hessian according

$$\underline{\gamma}(k) = -(\nabla^2 E(\underline{p}(k)) + \lambda(k) \cdot \underline{I})^{-1}, \quad \lambda(k) > 0 \quad (11)$$

where  $\underline{I}$  is the unit matrix. By incrementing or decrementing of  $\lambda(k)$  depending on the success are effected an error driven shifting between Newton- and Gradient-Method.

The calculation of the Hessian of the objective function can be done by further derivation of (6). According to the product rule is got

$$\begin{aligned} \frac{\partial^2 E(\underline{p})}{\partial p_\eta \cdot \partial p_n} &= \sum_{s=1}^S \frac{\partial^2 \hat{y}_s(\underline{p})}{\partial p_\eta \cdot \partial p_n} \cdot (\hat{y}_s(\underline{p}) - y_s) \\ &+ \sum_{s=1}^S \frac{\partial \hat{y}_s(\underline{p})}{\partial p_n} \cdot \frac{\partial \hat{y}_s(\underline{p})}{\partial p_\eta} \quad \forall n, \eta = 1, \dots, N \end{aligned} \quad (12)$$

as exact second derivation of the error function. According to the Gauß-Newton approximation the

second summand can be neglected. By means of the Jacobian matrix the calculation of the Hessian may be done by

$$\nabla^2 E(\underline{p}) \approx \underline{\hat{Y}}'^T(\underline{p}) \cdot \underline{\hat{Y}}'(\underline{p}) \quad (13)$$

The mentioned numerical methods can be used in the parameter-nonlinear case as local methods within the evolutionary parameter search of chapter 2.

The explanations of chapter 5 can always be used for an arbitrary classifier structure. The analytic calculation of necessary Jacobian matrix requires a concrete structure. The emulation of a nonlinear behaviour could be done i.e. by a polynom. Thereby the optimal parameters could be analytically calculated by (9). Disadvantageous is the large number of parameters in case of multi-input. For that a better structure is the Multilayer Perceptron considered in the next chapter.

## 6. CLASSIFICATION BY A MULTILAYER PERCEPTRON

As classifier be used a Multilayer Perceptron (MLP), which consist of three layer with  $I$  input units,  $H$  hidden units and in our case of one output unit. The transfer function of input- and output units is the identity and of the hidden units the hyperbolic tangent. Input- and hidden layer have additionally a bias unit with accordingly weights  $w_{I+1,h}^{(1)}$  and  $w_{H+1}^{(2)}$ .

Thus the output of the net can be calculated as

$$\hat{y}(\underline{u}) = \sum_{h=1}^H w_h^{(2)} \cdot \tanh\left(\sum_{i=1}^I w_{i,h}^{(1)} \cdot u_i + w_{I+1,h}^{(1)}\right) + w_{H+1}^{(2)} \quad (14)$$

By means of Taylor series it is easy to comprehend, that (14) can be transferred to a polynom. Thus the used net is able to emulate nonlinearities of arbitrary orders [4]. The parameters of the system are all weights in (14). The output is differentiable to the parameters, whereas the relation is nonlinear. Therefore gradient-based methods are useable. The necessary Jacobian matrix of the net output to the weights for all  $S$  training examples is

$$\underline{\hat{Y}}'(\underline{p}) = \begin{pmatrix} \frac{\partial \hat{y}_1}{\partial w_1^{(2)}} & \dots & \frac{\partial \hat{y}_1}{\partial w_{H+1}^{(2)}} & \frac{\partial \hat{y}_1}{\partial w_{11}^{(1)}} & \dots & \frac{\partial \hat{y}_1}{\partial w_{I+1,H}^{(1)}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \hat{y}_S}{\partial w_1^{(2)}} & \dots & \frac{\partial \hat{y}_S}{\partial w_{H+1}^{(2)}} & \frac{\partial \hat{y}_S}{\partial w_{11}^{(1)}} & \dots & \frac{\partial \hat{y}_S}{\partial w_{I+1,H}^{(1)}} \end{pmatrix} \quad (15)$$

The partial derivatives can be analytically calculated from (14). Built on knowledge from studies and applications, as optimization method be recommended the method of Marquardt-Levenberg (11). By means of the Jacobian matrix (15) can be calculated the gradient by (6) and the Hessian by (13).

As it is assumed by disturbed data, additionally to the training data are used validation data to also

calculate a net error  $E_v$ . To avoid overfitting, the optimization process is terminated, when  $E_v$  begins to increase [4].

Because of the multi-modal characteristic of the objective function  $E(p)$  the parameter optimization is performed by the hybrid algorithm of chapter 2. Thereby Marquardt-Levenberg is used as local method. The several networks of a population are different in its initial weights. The termination of the whole parameter search is done, if within a determined number of loops no improvement of  $E_v$  was reached.

Now it should be shown, how the whole procedure of chapter 3 is used for this task. System specific structure parameters are in the case of the three-layer MLP only the number of hidden units  $H$ . The number of input units are the number of relevant features. In preliminary studies was created a ranking list of the supplied features. For that was determinant the number of right recognized objects, when the feature was used separately in a linear classification. Thus the feature selection is reduced to determination of a number  $M$  for the  $M$  first features in the ranking list. Therefore for the structure variables is established a two-dimensional vector by  $(H, M)^T \in N^2$ . The possibilities for the incremental structure variation are:

- only improve of hidden units  $(H+1, M)^T$ ,
- only improve of features  $(H, M+1)^T$ ,
- improve of hidden units and features  $(H+1, M+1)^T$ .

Thus three subpopulations are generated in each generation of structure evolution. The selection of the best structure is based on a further set of test images, which are used to calculate the error rates, described in the next chapter.

## 7. RESULT AND CONCLUSION

The described procedure was successfully applied for several scan devices. The best results were reached using 10 features and 5 to 15 hidden units. For evaluation of the results are used three error rates, which are calculated on the basis of the test dataset.

- False Acceptance Rate (FAR) is the number of false classified imitations per the number of imitations.
- False Return Rate (FRR) is the number of false classified living objects per the number of living objects.
- Failure Rate (FR) is the number of false classified objects per the number of objects.

The error rates depend on the threshold of binarization. If the output of the classifier is under this threshold, then the object is classified as imitation, otherwise as alive. A typical result is represented in Figure 5. The threshold can be adjusted

by the user on the basis of this diagram. It is to see, there is a remaining error depending on the threshold.

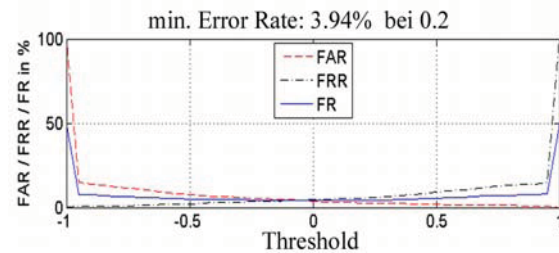


Figure 5 Example of resulting Error Curves

In this paper should be shown, that the simulated evolution can be used to support design and optimization of systems. Advantageous is the relative simple applicability with less problem specific knowledge. Disadvantageous is the slow and not assured local convergence compared with classical optimization methods. A good symbiosis can be generated from classical and evolutionary methods by hybrid EA. Based on knowledge from various researches and own analyses are proposed an algorithmic concept for that with regard to the systems engineering. This can expediently be used for image recognition, because hereby the trials of the simulated evolution are not linked with time-consuming simulations. At present the proposed methods are evaluated and partial improved. Therefore a MATLAB®-Tool was developed, which enables the configuration and parametrization of hybrid algorithms and their experimental analyzation.

## 8. REFERENCES

- [1] H. Pohlheim: „Evolutionäre Algorithmen Verfahren, Operatoren und Hinweise für die Praxis“, Fachbuch, Springer Verlag Berlin Heidelberg New York, 2000
- [2] W. Jacob: „Eine neue Methodik zur Erhöhung der Leistungsfähigkeit Evolutionärer Algorithmen durch die Integration lokaler Suchverfahren“, Dissertation, Universität Karlsruhe, 2004
- [3] R. Hooke, T.A. Jeeves: „Direct search solution of numerical and statistical problems“, Journal of the ACM 8, 1961
- [4] P. Otto: „Identifikation nichtlinearer Systeme mit Künstlichen Neuronalen Netzen“, at-Automatisierungstechnik 43 (1995)